Amendment to the Claims

1.   (currently amended)   A method of transferring a packet to a host computer
system, wherein the packet is received at a communication device from a network,
comprising:

parsing a header portion of a first packet received at a network interface
for the host computer system to determine if said first packet conforms to a ~~pre-selected~~
TCP protocol;

generating a flow key to identify a first communication flow that includes
said first packet, wherein said flow key includes a TCP connection for the
communication flow;

associating an operation code with said first packet, wherein said
operation code indicates a status of said first packet, including whether said packet is a
candidate for transfer to the host computer system that avoids processing said header
portion by the host computer system in accordance with said ~~pre-selected~~ TCP protocol;
and

processing, by the network interface, said packet ~~the~~ according to the TCP
connection, including updating a control block representing the TCP connection on the
network interface.

2.   (previously presented)   The method of claim 1, wherein said parsing
comprises:

copying the header portion of said first packet into a header memory,
wherein said header memory does not store a remainder of said first packet; and

examining said header portion according to a series of parsing
instructions;

wherein said parsing instructions are configured to reflect a set of pre-
selected communication protocols.

3.   (original)   The method of claim 2, wherein said parsing instructions are
updateable.

4.   (original)   The method of claim 2, further comprising copying a value from a field in a header of said header portion.

5.   (previously presented)   The method of claim 1, wherein said parsing comprises:

        extracting an identifier of a source of said first packet from said header portion; and

        extracting an identifier of a destination of said first packet from said header portion, wherein said destination is an application running on the host computer system.

6.   (original)   The method of claim 5, wherein said generating comprises combining said source identifier and said destination identifier.

7.   (previously presented)   The method of claim 1, wherein said generating comprises retrieving an identifier of a communication connection for the host computer system from said header portion.

8.   (original)   The method of claim 1, further comprising storing said first packet in a packet memory prior to said transferring.

9.   (previously presented)   The method of claim 1, further comprising storing said flow key in a flow database, wherein said flow database is configured to facilitate TCP management of said first communication flow.

10.   (original)   The method of claim 9, further comprising associating a flow number with said first packet, wherein said flow number comprises an index of said flow key within said flow database.

11.   (original)   The method of claim 10, further comprising storing said flow number in a flow memory.

12.   (original)   The method of claim 9, further comprising updating an entry in said flow database associated with said flow key when a second packet in said first communication flow is received.

13.   (currently amended)   A non-transitory computer readable storage medium storing instructions that, when executed by a computer, cause the computer to perform a method of transferring a packet received at a network interface from a network to a host computer system, the method comprising:

receiving a packet from a network at a network interface for a host computer system;

parsing a header portion of said packet to extract an identifier of a source entity and an identifier of a destination entity;

generating a flow key from said source identifier and said destination identifier to identify a communication flow comprising said packet, wherein said flow key ~~includes~~ identifies a TCP connection for the communication flow;

determining whether a header in said header portion conforms to ~~a pre-selected~~ the TCP protocol;

storing said flow key in a database;

associating an operation code with said packet, wherein said operation code identifies a status of said packet, including whether said packet is to be TCP processed by the host computer;

storing said packet in a packet memory;

if said ~~header conforms to said pre-selected protocol~~ operation code indicates that said packet is not to be TCP processed by the host computer:

storing a data portion of said packet in a re-assembly buffer; and

storing said header portion in a header buffer; and

processing, by the network interface, said packet ~~the~~ according to the TCP connection, including updating a control block representing the TCP connection on the network interface.

14. (original) The method of claim 1, wherein said associating comprises:

retrieving one or more header fields of said header portion; and analyzing said header fields to determine said status of said first packet.

15. (original) The method of claim 14, wherein said analyzing comprises:

determining whether said first packet includes a data portion; and

if said first packet includes a data portion, determining whether said data portion exceeds a pre-determined size.

16. (original) The method of claim 14, wherein said analyzing comprises determining whether said first packet was received out of order in said first communication flow.

17. (original) The method of claim 1, further comprising storing said operation code in a control memory.

18. (currently amended) The method of claim 1, wherein said first packet is determined to conform to said ~~pre-selected~~ TCP protocol, said transferring comprising:

storing a data portion of said first packet in a re-assembly storage area, wherein said re-assembly storage area is configured to only store data portions of packets in said first communication flow; and

storing one or more headers from said header portion in a header storage area.

19. (original) The method of claim 1, wherein said transferring comprises:

if said first packet is smaller than a predetermined threshold, storing said first packet in a first storage area; and

if said first packet is larger than said predetermined threshold, storing said first packet in a second storage area.

20.   (original)   The method of claim 1, further comprising determining whether a second packet received from said network is part of said first communication flow.

21.   (original)   The method of claim 20, wherein said determining comprises:

maintaining a packet memory configured to store one or more packets received from said network;

maintaining a flow memory configured to store, for each of said one or more packets, an identifier of a communication flow comprising said packet; and

searching said flow memory for a first identifier of said first communication flow.

22.   (original)   The method of claim 21, wherein said first identifier comprises said flow key.

23.   (original)   The method of claim 21, wherein said first identifier comprises a flow number of said first packet, wherein said flow number is an index of said flow key within a flow database.

24.   (currently amended)   A method of transferring a packet to a host computer system, wherein the packet is received at a communication device for the host computer system from a network, comprising:

parsing a header portion of a first packet received at a communication device to determine if said first packet conforms to a ~~pre-selected~~ TCP protocol, wherein said header portion includes an IP address of the host computer system;

generating a TCP connection to identify a first communication flow that includes said first packet;

associating a summary with said first packet, wherein said summary indicates a status of said first packet, including whether said packet is a candidate for

transfer to the host computer system that avoids processing <u>said header portion by the host computer system</u> in accordance with said ~~pre-selected~~ <u>TCP</u> protocol; and

processing, by the network interface, said packet ~~the~~ according to the TCP connection, <u>including updating a control block representing the TCP connection on the network interface.</u>

25.   (previously presented)   The method of claim 24, wherein said generating a TCP connection is performed by said host computer system, and further comprising:

transferring said connection from said host computer system to said communication device.

26.   (previously presented)   The method of claim 24, further comprising:

receiving a second packet from a second communication flow; and

processing said second packet by said communication device in accordance with said connection.

27.   (original)   The method of claim 1, further comprising alerting said host computer system to the arrival of said first packet.

28.   (original)   The method of claim 1, further comprising:

maintaining a packet memory configured to store packets received from said network; and

randomly discarding a packet if said packet memory contains a pre-determined level of traffic.

29.   (original)   The method of claim 1, wherein said parsing includes determining, by hardware of the communication device, a session layer protocol of the header.

30.   canceled.

31.   (original)   The method of claim 1, wherein said communication device is a network interface.

32.   (currently amended)   A method of transferring a packet received at a network interface of ~~to~~ a host computer system, comprising:

> receiving, by the network interface, a packet from a network;
>
> storing, by the network interface, said packet in a packet memory;
>
> parsing, by the network interface, a header portion of said packet;
>
> extracting, by the network interface, a value stored in said header portion;
>
> identifying, by the network interface, a communication flow comprising said packet, wherein said communication flow key ~~includes~~ identifies a TCP connection and a first hop medium access control (MAC) layer address;
>
> determining, by the network interface, whether a header in said header portion conforms to ~~a pre-selected~~ the TCP protocol;
>
> determining, by the network interface, whether a second packet in said packet memory is part of said communication flow;
>
> processing, by the network interface, said packet ~~the~~ according to the TCP connection, including updating a control block representing the TCP connection on the network interface;
>
> if the host computer system contains a plurality of processors such that a first of the processors is on the network interface and a second of the processors is on the host computer, identifying a processor of the first and second processors to process said second packet; and
>
> storing, by the network interface, said packet in a host memory area.

33.   (currently amended)   A method of transferring a packet received at a network interface from a network to a host computer system, comprising:

> receiving a packet from a network at a network interface ~~for~~ of a host computer system;
>
> parsing a header portion of said packet to extract an identifier of a source entity and an identifier of a destination entity;

generating a flow key from said source identifier and said destination identifier to identify a communication flow comprising said packet, wherein said flow key includes a TCP connection for the communication flow and a first hop medium access control (MAC) layer address;

determining whether a header in said header portion conforms to a pre-selected protocol;

storing said flow key in a database;

associating an operation code with said packet, wherein said operation code identifies a status of said packet;

storing said packet in a packet memory;

if said header conforms to ~~said pre-selected~~ the TCP protocol:

storing a data portion of said packet in a re-assembly buffer;

storing said header portion in a header buffer; and

processing, by the network interface, said packet ~~the~~ according to the TCP connection.

34. (original)   The method of claim 33, wherein said parsing comprises executing a series of updateable instructions configured to parse a packet header conforming to one of a set of pre-selected protocols.

35. (original)   The method of claim 33, further comprising storing said operation code in a control memory.

36. (original)   The method of claim 33, further comprising storing a flow number of said packet in a flow memory, wherein said flow number comprises an index of said flow key in said database.

37. (original)   The method of claim 36, further comprising indicating whether said packet memory includes another packet with said flow number or said flow key.

38.   (currently amended)   The method of claim 33, wherein the host computer system comprises multiple processors, ~~further comprising identifying a first processor in the host computer system to process said packet in accordance with said pre-selected protocol~~ for processing network packets in accordance with the TCP protocol.

39.   (currently amended)   The method of claim 38, further comprising:

receiving a second packet at said network interface, wherein said second packet is part of a second communication flow; and

identifying a ~~second~~ processor in the host computer system to process said second packet.

40.   (original)   The method of claim 33, further comprising informing said host computer system of said receipt of said packet.

41.   (original)   The method of claim 1, wherein said transferring said first packet to a host computer system comprises:

analyzing a session layer part of said header to allocate a list of memory addresses of the host computer for storing data from a plurality of packets that correspond to said communication flow.

42.   (currently amended)   An apparatus for transferring a packet to a host computer system, comprising:

a traffic classifier, disposed in a network interface for the host computer system, configured to classify a first packet received from a network by a communication flow that includes said first packet;

a packet memory, disposed in the network interface, configured to store said first packet;

a packet batching module, disposed in the network interface, configured to determine whether another packet in said packet memory belongs to said communication flow;

a flow re-assembler, disposed in the network interface, configured to re-assemble a data portion of said first packet with a data portion of a second packet in said communication flow; and

a processor, disposed in the network interface, that maintains a TCP connection for the communication flow, the TCP connection stored as a control block on the network interface.

43.   (original)   The apparatus of claim 42, wherein said traffic classifier comprises:

a parser configured to parse a header portion of said first packet;

a flow database configured to store a flow key identifying said communication flow; and

a flow database manager configured to manage said flow database;

wherein said flow key is generated from an identifier of a source of said first packet and an identifier of a destination of said first packet.

44.   (currently amended)   A computer system for receiving a packet from a network, comprising:

a memory configured to store packets received from a network; and

a network interface for the computer system, the network interface configured to receive a first packet from said network, the network interface comprising:

a parser configured to extract information from a header portion of a first packet;

a flow manager configured to examine said information;

a flow database configured to store an identifier of a first communication flow comprising multiple packets, including said first packet; and

a re-assembler for storing data portions of said multiple packets without header portions in a first portion of said memory; and

a processor for processing said first packet and for maintaining a TCP connection for the communication flow, the TCP connection stored as a control block on the network interface.

45.   (previously presented)   The apparatus of claim 42, further comprising:

a load distributor for identifying a first processor within the host computer system for processing said first packet and said second packet;

wherein said load distributor identifies a second processor in the host computer system for processing a packet from a different communication flow, wherein said first processor is a part of said network interface and said second processor is a part of said host computer.

46.   (original)   The method of claim 1, wherein said operation code indicates whether the packet corresponds to Transport Control Protocol (TCP).

47.   (currently amended)   A device for receiving a packet from a network and transferring the packet to a host computer system, comprising:

a parser configured to parse a header portion of a packet received from a network, wherein said parsing comprises:

determining whether a header within said header portion conforms to one of a set of communication protocols; and

if said header conforms to one of said communication protocols, extracting information from said header portion to identify a communication flow to which said packet belongs;

a flow memory configured to store a flow identifier for identifying said communication flow;

a flow manager configured to assign an operation code to said packet, wherein said operation code:

indicates a status of said packet; and

indicates a manner of transferring said packet to the host computer system;

a packet memory configured to store said packet; and

a transfer module configured to transfer said packet from said packet memory to the host computer system in accordance with said operation code, wherein said device is a peripheral device for the host computer system;

wherein said device maintains a TCP connection for the communication flow, the TCP connection stored as a control block on the device.

48.   (original)   The device of claim 47, wherein the device is a network interface.

49.   (original)   The device of claim 47, said flow memory comprising a flow database configured to store a flow key, wherein said flow key is assembled from an identifier of a source of said packet and an identifier of a destination of said packet.

50.   (original)   The device of claim 47, wherein said flow manager is further configured to update said flow memory as additional packets in said communication flow are received from the network.

51.   (original)   The device of claim 47, said flow memory comprising a flow memory configured to store a flow number, wherein said flow number comprises an index of said communication flow in a flow database.

52.   (original)   The device of claim 47, further comprising a control memory configured to store said operation code.

53.   (currently amended)   A non-transitory computer readable storage medium storing instructions that, when executed by a computer, cause the computer to perform a method of transferring a packet from a communication device for a host computer to the host computer, the method comprising:

parsing a header portion of a first packet received at a network interface for the host computer to determine if said first packet conforms to a ~~pre-selected~~ TCP protocol;

generating a flow key to identify a first communication flow that includes said first packet, wherein said flow key includes an IP address and a TCP port for a TCP connection of the host computer system;

associating an operation code with said first packet, wherein said operation code indicates a status of said first packet, including whether said packet is a candidate for transfer to the host computer system that avoids processing said header portion by the host computer system in accordance with said ~~pre-selected~~ TCP protocol; and

processing, by the network interface, said packet ~~the~~ according to the TCP connection, including updating a control block representing the TCP connection on the network interface.

54.   canceled.

55.   (original)   The device of claim 47, wherein said transfer module is configured to transfer a data portion of said packet into one of a set of host memory areas in accordance with said operation code.

56.   (original)   The device of claim 47, further comprising a packet batching module configured to determine whether said packet memory contains another packet in said communication flow.

57.   (original)   The method of claim 1, further comprising:

receiving, by said communication device, a second packet that corresponds to said communication flow, said second packet including at least a transport layer header and data; and

transferring said data from said communication device to said host computer system, without transferring said transport layer header from said communication device to said host computer system.

58.   (previously presented)   An apparatus for transferring a packet from a network to a host computer system, comprising:

a parser module configured to:

parse a header portion of a first packet received from a network to extract an identifier of a source of said first packet and an identifier of a destination of said first packet, wherein said destination is part of the host computer system;

generate a flow key from said source identifier and said destination identifier to identify a communication flow comprising said first packet; and

determine whether a header in said header portion conforms to a pre-selected protocol;

a flow database configured to store said flow key;

a flow database manager configured to associate an operation code with said first packet, wherein said operation code identifies a status of said first packet;

a packet memory configured to store said first packet; and

a transfer module configured to:

if said header conforms to said pre-selected protocol:

store a data portion of said first packet in a re-assembly buffer; and

store said header portion in a header buffer; and

if said header conforms to a protocol other than said pre-selected protocol, store said packet in a non-re-assembly buffer;

wherein said apparatus maintains a TCP connection for the communication flow.

59.   (original)   The apparatus of claim 58, wherein said transfer module comprises a re-assembly engine configured to re-assemble, in said re-assembly buffer, a data portion of said first packet with a data portion of a second packet in said first communication flow.

60.   (original)   The apparatus of claim 58, further comprising a flow memory configured to store a flow number associated with said first packet, wherein said flow number comprises an index of said flow key in said flow database.

61.   (original)   The apparatus of claim 58, further comprising:

a load distributor configured to identify a first processor in said host computer system for processing said first packet, said first processor being identified on the basis of said flow key;

wherein said host computer system is a multi-processor computer system; and

wherein a second processor in said host computer system is identified for processing a packet from a communication flow other than said first communication flow.

62.   (original)   The apparatus of claim 58, further comprising:

a packet batching module configured to determine whether said packet memory includes another packet in said first communication flow.